

### **AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A "Device for Reducing [[the]]\_a Width of Graph" which reduces [[the]]\_a width of [[the]]\_a Binary Decision Diagram for Characteristic Function (BDD\_for\_CF), where BDD\_for\_CF is a characteristic function  $\chi(X, Y)$  defined in Equation (1),  $X=(x_1, \dots, x_n)$  ( $n \in N$ ,  $N$  is a set of natural numbers) denotes input variables,  $Y=(y_0, \dots, y_{m-1})$  ( $m \geq 2$ ,  $m \in N$ ) denotes [[the]] output variables of a multiple-output logic function  $F(X)$ , and  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  is an incompletely specified function to [[the]]\_an output including don't care, said device comprising:

(A) "Means to Store Node Table" storing the node table which is [[the]]\_a table of node data that consists of [[the]] labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , where the labels of variables are labels given to [[the]] variables  $z_i$  ( $z_i \in (X \cup Y)$ ) corresponding to [[said]] each non-terminal node  $v_i$  in the BDD\_for\_CF of the multiple-output logic function  $F(X)$ , and a pair of edges

$e_0(v_i)$  and  $e_1(v_i)$  that points [[the]]\_to next transition child node(s) when [[the]] input values of  $z_i$  ( $z_i \in (X \cup Y)$ ) are 0 and 1;

(B) "Means to Find [[the]] Dividing Lines" setting [[the]]\_a height of the partition  $lev$  which partitions BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table";

(C) "Means to Generate Column Functions" generating a column function which represents [[the]]\_a column of [[the]]\_a decomposition chart derived by [[the]]\_a functional decomposition from said node table stored in said "Means to Store Node Table", where the

decomposition is obtained by partitioning said BDD\_for\_CF by said height of the partition /lev set by said "Means to Find the Dividing Lines"; and

(D) "Means to Reconstruct Assigned BDD" which assigns [[the]] constants to [[the]] don't care in [[the]] compatible column functions of column function generated by said "Means to Generate Column Functions", and consequently [[makes]] assigns these compatible column functions to the identical column functions (~~hereafter, assigned column functions~~), and reconstructs said BDD\_for\_CF using [[the]] new assigned column function, and finally updates the node table in said "Means to Store Node Table", wherein Equation (1) is defined as follows:

[Equation 1]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} \{ \bar{y}_i f_{i_0} \vee y_i f_{i_1} \vee f_{i_d} \} \quad (1)$$

where  $f_{i_0}$ ,  $f_{i_1}$ ,  $f_{i_d}$  are [[the]] OFF function, the ON function and [[the]] DC function defined in Equation (2), respectively, wherein Equation (2) is defined as follows:

[Equation 2]

$$f_{i_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$

2. (Currently Amended) The "Device for Reducing [[the]] a Width of Graph" according to Claim 1, wherein:

the device further comprises;

(E) "Means to Store Compatible Graphs" storing [[the]] compatible graph as a table of function node data, [[that]] which is a table of column function labels of [[said]] each function node and [[the]] data of compatible edges connected to the function node, where the compatible graph is a graph which has said column functions as nodes (function

nodes), and wherein a pair of function nodes corresponding to the column functions compatible with each other are connected by an edge (compatible edge);

(F) "Means to Generate Compatible Edges" which selects [[the]]\_a pair of compatible column function from [[the]]\_a set of column functions corresponding to said each function node data, stored in said "Means to Store Compatible Graphs", and then adds a compatible edge which connects these function nodes with function node data corresponding to these compatible column functions, and finally updates the function node data stored in said "Means to Store Compatible Graphs"; and

(G) "Means to Generate Cliques" covering nodes with [[the]]\_a minimum number of complete subgraphs (cliques) for all nodes in said compatible graph, and then generating clique data of function node set contained within the clique; and

wherein said "Means to Generate Column Functions" further generates column functions corresponding to each edge of nodes at said height of the partition /ev set by said "Means to Find the Dividing Lines" from said node table stored in "Means to Store Node Table", and then generates said function node data having column function labels corresponding to these column functions, and then stores in said "Means to Store Compatible Graphs", and

wherein said "Means to Reconstruct Assigned BDD" further reconstructs said BDD\_for\_CF by making some column functions [[to]]\_into the identical assigned column functions by assigning the constants to don't care of the column functions corresponding to each function node contained in the clique data produced by said "Means to Generate Cliques", and updates said node table in said "Means to Store Node Table".

3. (Currently Amended) The "Device for Reducing [[the]]\_a Width of Graph" according to Claim 1, wherein:

said "Means to Find the Dividing Lines" further sets the height of the partition  $lev$  sequentially from the height of the child node of [[the]]\_a root node in BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table", towards [[the]]\_a low height, and

said "Means to Reconstruct Assigned BDD" further reconstructs sequentially in said each height of the partition  $lev$  set by said "Means to Find the Dividing Lines".

4. (Currently Amended) A "Device for Logic Synthesis" which generates look-up tables (LUTs) of [[the]] data for constructing logic circuits corresponding to [[said]]\_a multiple-output logic function  $F(X)$  from the BDD\_for\_CF (Binary Decision Diagram for Characteristic Function) of the multiple-output logic function  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  with input variables  $X=(x_1, \dots, x_n)$  ( $n \in N$ ), said device comprising:

(A) "Means to Store Node Table" storing BDD\_for\_CF representing [[the]]\_a characteristic function  $\chi(X, Y)$  (where  $Y=(y_0, \dots, y_{m-1})$  ( $m \geq 2, m \in N$ ) denotes output variables of  $F(X)$ ) defined in Equation (3), as [[the]]\_a node table which is [[the]]\_a table of node data that consists of [[the]] labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , where said multiple-output logic function  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  is a completely specified function, the labels of variables are labels given to [[the]] variables  $z_i$  ( $z_i \in (X \cup Y)$ ) corresponding to [[said]] each non-terminal node  $v_i$  in the BDD\_for\_CF, and a pair of edges  $e_0(v_i)$  and  $e_1(v_i)$

that points [[the]] to next transition child nodes when [[the]] input values of  $z_i$  ( $z_i \in (X \cup Y)$ ) are 0 and 1;

(B) "Means to Store LUTs" storing said LUTs;

(C) "Means to Find [[the]] Dividing Lines" setting [[the]] a height of the partition  $lev$  which partitions BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table";

(D) "Means to Reduce by Shorting" executing shorten-processing [[that]] which is [[the]] a processing to replace [[the]] edge  $e_c(v_k)$  that points [[the]] to a node  $v_j$  among two edges  $e_0(v_k)$  and  $e_1(v_k)$  of [[the]] a parent node  $v_k$  of the node  $v_j$ , by [[the]] edge  $e_b(v_j)$  other than [[the]] edge  $e_a(v_j)$  of the node  $v_j$ , in the case that [[the]] a terminal node related to  $\chi(X, Y)=0$  pointed by the edge  $e_a(v_j)$  of either [[the]] edges  $e_0(v_j)$  or  $e_1(v_j)$  of the node  $v_j$ , about the node data of the node  $v_j$  related to [[the]] a variable  $y_i (\in Y)$  representing output and the parent node  $v_k$  of the node  $v_j$ , where the nodes  $v_j$  and  $v_k$  are contained in [[the]] a subgraph  $B_0$  including [[the]] a root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD\_for\_CF to [[the]] two subgraphs  $B_0$  and  $B_1$  at [[the]] a partition line in said height of the partition  $lev$ ;

(E) "Means to Measure [[the]] a Width of BDDs" which counts [[the]] a number of the edges that point [[the]] to child nodes of the non-terminal nodes, whose height is smaller than said height of the partition  $lev$ , among the edges which are the non-terminal nodes in BDD\_for\_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied and which belong to the non-terminal nodes whose height is larger than said height of the partition  $lev$  (where the edges pointing to the same node is counted as one, and the

edge pointing to [[point the]]\_a constant 0 is disregarded), and produces the number of the edges as [[the]]\_a width  $W$  at the partition line in said height of the partition  $lev$ ;

(F) "Means to Compute the Intermediate Variables" calculating the number of intermediate variables  $u$  following Equation (4), using the width  $W$  produced by said "Mean to Measure the Width of BDDs";

(G) "Means to Generate LUTs" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph  $B_0$  including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said BDD\_for\_CF into two subgraphs at the partition line in said height of the partition  $lev$ ; and

(H) "Means to Re-construct BDDs" which generates a binary tree which has the same number of control inputs as the number of intermediate variables  $u$  which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the BDD\_for\_CF by replacing the node data of non-terminal nodes in subgraph  $B_0$  of BDD\_for\_CF stored in said "Means to Store Node Table" by the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD\_for\_CF, wherein Equation (3) and Equation (4) are defined as follows:

[Equation 3]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \quad (3)$$

[Equation 4]

$$u = \lceil \log_2 W \rceil \quad (4)$$

5. (Currently Amended) The "Device for Logic Synthesis" according to Claim 4, wherein said "Means to Store Node Table" stores the BDD\_for\_CF as a node table, where said BDD\_for\_CF is a graph that represents the characteristic function  $\chi(X, Y)$  (where  $Y = (y_0, \dots, y_{m-1})$  ( $m \geq 2, m \in N$ ) denotes the output variables of  $F(X)$ ) defined in Equation (5), with said multiple-output logic function  $F(X) = (f_0(X), \dots, f_{m-1}(X))$  of an incompletely specified function that includes don't cares in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , said labels of variables are labels given to the variables  $z_i$  ( $z_i \in (XUY)$ ) corresponding to said each non-terminal node  $v_i$  in the BDD\_for\_CF, and said pair of edges

$e_0(v_i)$  and  $e_1(v_i)$  that points to the next transition child nodes when the values of  $z_i$  ( $z_i \in (XUY)$ ) are 0 and, wherein Equation (5) is defined as follows:

[Equation 5]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} \{ \bar{y}_i f_{i,0} \vee y_i f_{i,1} \vee f_{i,d} \} \quad (5)$$

where  $f_{i,0}, f_{i,1}, f_{i,d}$  are [[the]] OFF function, [[the]] ON function and [[the]] DC function defined in Equation (6), respectively, wherein Equation (6) is defined as follows:

[Equation 6]

$$f_{i,0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i,1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i,d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

6. (Currently Amended) A "Device for Logic Synthesis" which generates look-up tables (LUTs) of the data for constructing logic circuits corresponding to [[said]] multiple-output logic function  $f(X)$  from [[the]] a BDD\_for\_CF (Binary Decision Diagram for

Characteristic Function) of the multiple-output logic function  $F(X)=F_0(X), \dots, f_{m-1}(X)$  with input variables  $X=(x_1, \dots, x_n)$  ( $n \in N$ ), said device comprising:

(A) "Means to Store Node Table" storing BDD\_for\_CF representing [[the]]\_a characteristic function  $x(X, Y)$  (where  $Y=(y_0, \dots, y_{m-1})$  ( $m \geq 2, m \leq N$ ) denotes output variables of  $F(X)$ ) defined in Equation (3), as [[the]]\_a node table which is [[the]]\_a table of node data that consists of [[the]] labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , where said multiple-output logic function  $F(X)=f_0(X), \dots, f_{m-1}(X)$  is a completely specified function, the labels of variables are labels given to the variables  $z_i$  ( $z_i \in (X \cup Y)$ ) corresponding to [[said]] each non-terminal node  $v_i$  in the BDD\_for\_CF, and a pair of edges  $e_0(v_i)$  and  $e_1(v_i)$  that points [[the]]\_to next transition child nodes when [[the]] input values of  $z_i$  ( $z_i \in (X \cup Y)$ ) are 0 and 1;

(B) "Means to Store LUTs" storing said LUTs;

(C) "Means to Find [[the]] Dividing Lines" setting [[the]]\_a height of the partition  $lev$  which partitions BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table";

(D) "Means to Reduce by Shorting" executing shorten-processing [[that]]\_which is [[the]]\_a processing to replace [[the]] edge  $e_c(v_k)$  that points [[the]]\_to a node  $v_j$  among two edges  $e_0(v_k)$  and  $e_1(v_k)$  of [[the]]\_a parent node  $v_k$  of the node  $v_j$ , by [[the]]\_an edge  $e_b(v_j)$  other than [[the]]\_an edge  $e_a(v_j)$  of the node  $v_j$ , in the case that [[the]]\_a terminal node related to  $\chi(X, Y)=0$  pointed by the edge  $e_a(v_j)$  of either the edges  $e_0(v_j)$  or  $e_1(v_j)$  of the node  $v_j$ , about the node data of the node  $v_j$  related to [[the]]\_a variable  $y_i$  ( $i \in Y$ ) representing output



and the parent node  $v_k$  of the node  $v_j$ , where the nodes  $v_j$  and  $v_k$  are contained in [[the]]\_a subgraph  $B_0$  including [[the]]\_a root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD\_for\_CF to [[the]] two subgraphs  $B_0$  and  $B_1$  at [[the]]\_a partition line in said height of the partition  $lev$ ;

(E) "Means to Measure [[the]] Width of BDDs" which counts the number of [[the]] edges that point [[the]]\_to child nodes of [[the]] non-terminal nodes, whose height is smaller than said height of the partition  $lev$ , among the edges which are the non-terminal nodes in BDD\_for\_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied and which belong to the non-terminal nodes whose height is larger than said height of the partition  $lev$  (where the edges pointing to the same node is counted as one, and the edge to point [[the]]\_to\_a constant 0 is disregarded), and produces the number of edges as [[the]]\_a width  $W$  at the partition line in said height of the partition  $lev$ ;

(F) "Means to Compute [[the]] Intermediate Variables" calculating the number of intermediate variables  $u$  following Equation (4), using the width  $W$  produced by said "Mean to Measure the Width of BDDs";

(G) "Means to Generate LUTs" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph  $B_0$  including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said BDD\_for\_CF into two subgraphs at the partition line in said height of the partition  $lev$ ; and

(H) "Means to Re-construct BDDs" which generates a binary tree which has the same number of control inputs as the number of intermediate variables  $u$  which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the

BDD\_for\_CF by replacing the node data of non-terminal nodes in subgraph  $B_0$  of BDD\_for\_CF stored in said "Means to Store Node Table" by the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD\_for\_CF, wherein Equation (3) and Equation (4) are defined as follows:

[Equation 3]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \quad (3)$$

[Equation 4]

$$u = \lceil \log_2 W \rceil \quad (4)$$

wherein said "Means to Store Node Table" stores the BDD\_for\_CF as a node table, where said BDD\_for\_CF is a graph that represents the characteristic function  $\chi(X, Y)$  (where  $Y = (y_0, \dots, y_{m-1})$  ( $m \geq 2$ ,  $m \in \mathbb{N}$ ) denotes [[the]] output variables of  $F(X)$ ) defined in Equation (5), with said multiple-output logic function  $F(X) = (f_0(X), \dots, f_{m-1}(X))$  of an incompletely specified function that includes don't cares in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , said labels of variables are labels given to the variables  $z_i$  ( $z_i \in (XUY)$ ) corresponding to said each non-terminal node  $v_i$  in the BDD\_for\_CF, and said pair of edges  $e_0(v_i)$  and  $e_1(v_i)$  that points to the next transition child nodes when the values of  $z_i$  ( $z_i \in (XUY)$ ) are 0 and, wherein Equation (5) is defined as follows:

[Equation 5]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i_0} \vee y_i f_{i_1} \vee f_{i_d}\} \quad (5)$$

where  $f_{i_0}, f_{i_1}, f_{i_d}$  are ~~[[the]]~~ OFF function, ~~[[the]]~~ ON function and ~~[[the]]~~ DC function defined in Equation (6), respectively, wherein Equation (6) is defined as follows:

[Equation 6]

$$f_{i_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

~~the device comprises the "Device for Reducing the Width of Graph" according to claim 1, and~~

said "Means to Reduce by Shorting" reduces the width of BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table" ~~by said "Device for Reducing the Width of Graph"~~, and then performs said shorten-processing on ~~[[the]]~~ updated node table.

7. (Currently Amended) The "Device for Logic Synthesis" according to claim 4, wherein the device further comprises:

"Means to Decide ~~[[the]]~~ Ordering of Output Variables" deciding ~~[[the]]~~ order  $\pi$  of elements of said multiple-output logic function  $F(X)$  to minimize ~~[[the]]~~ a value of  $T$  represented in Equation (7), where  $\pi = (\pi[0], \dots, \pi[m-1])$  ( $\pi[i]=j$  represents that  $f_j$  is the  $i$ 'th element) is the order of the logic functions  $f_0(X), \dots, f_{m-1}(X)$  that are elements of said multiple-output logic function  $F(X)$ , and  $\text{supp}(f_j)$  is ~~[[the]]~~ a set of input variables that influence the logic function  $f_j (\in F(X))$ ;

"Means to Decide [[the]] Ordering of all [[the]] Variables" deciding [[the]] an order of the variables  $y_j (\in Y)$  representing the outputs and input variables  $x_i (\in X)$  in the order  $P$  that satisfies Equation (8); and

"Means to Generate BDDs" which generates node data of the BDD\_for\_CF according to [[the]] an order  $P$  decided in said "Means to Decide the Ordering of all the Variables", and then stores in said "Means to Generate BDDs", wherein Equation (7) and Equation (8) are defined as follows:

[Equation 7]

$$T = \sum_{k=0}^{m-1} \left| \bigcup_{l=0}^k \text{supp}(f_{\pi[l]}) \right| \quad (7)$$

[Equation 8]

$$P = \left( \text{supp}(f_{\pi[0]}), y_{\pi[0]}, \text{supp}(f_{\pi[1]}) - \text{supp}(f_{\pi[0]}), y_{\pi[1]}, \text{supp}(f_{\pi[2]}) - \left( \sum_{k=0}^1 \text{supp}(f_{\pi[k]}) \right), y_{\pi[2]}, \right. \\ \left. \dots, \text{supp}(f_{\pi[m-1]}) - \left( \sum_{k=0}^{m-2} \text{supp}(f_{\pi[k]}) \right), y_{\pi[m-1]} \right) \quad (8)$$

8. (Currently Amended) A Method to Reduce [[the]] a Width of a Graph which reduces [[the]] a width of [[the]] a BDD\_for\_CF (Binary Decision Diagram for Characteristic Function), in [[the]] a system comprising "Means to Store Node Table" which stores [[the]] a node table which is [[the]] a table of node data that consists of the labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , where BDD\_for\_CF is a characteristic function  $\chi(X, Y)$  defined in Equation (9),  $X=(x_1, \dots, x_n)(n \in N, N$  is a set of natural numbers) are input variables,  $Y=(y_0, \dots, y_{m-1})(m \geq 2, m \in N)$  denotes [[the]] output variables of a multiple-output

logic function  $F(X)$ ,  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  is an incompletely specified function to [[the]] an output including don't care, the labels of variables are labels given to [[the]] variables  $z_i$  ( $z_i \in (XUY)$ ) corresponding to [[said]] non-terminal node  $v_i$  in the BDD\_for\_CF of the multiple-output logic function  $F(X)$ , and a pair of edges  $e_0(v_i)$  and  $e_1(v_i)$  that points [[the]] to next transition child nodes when [[the]] input values of  $z_i$  ( $z_i \in (XUY)$ ) are 0 and 1, the method comprising the steps of:

a) a "Step to Find Dividing Lines" setting [[the]] a height of the partition  $lev$  which partitions BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table";

b) a "Step to Generate Column Functions" generating a column function which represents [[the]] a column of [[the]] a decomposition chart derived by [[the]] a functional decomposition from said node table stored in said node table in said "Means to Store Node Table", where the decomposition is obtained by partitioning said BDD\_for\_CF by said height of the partition  $lev$  set in said "Step to Find Dividing Lines"; and

c) a "Step to Reconstruct Assigned BDD", performed by a computer, assigning [[the]] constants to [[the]] don't care in [[the]] compatible column functions of the column function generated in said "Step to Generate Column Functions", and consequently making assigning these compatible column functions to the identical column functions (~~hereafter, assigned column functions~~), and reconstructing said BDD\_for\_CF using new assigned column functions, and finally updating the node table in said "Means to Store Node Table", wherein Equation (9) is defined as follows:

[Equation 9]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i_0} \vee y_i f_{i_1} \vee f_{i_d}\} \quad (9)$$

where  $f_{i_0}, f_{i_1}, f_{i_d}$  are [[the]] OFF function, [[the]] ON function and [[the]] DC function defined in Equation (10), respectively, wherein Equation (10) is defined as follows:  
[Equation 10]

$$f_{i_0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i_d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \quad (10)$$

9. (Currently Amended) The Method to Reduce the Width of Graph according to Claim 8 wherein:

said system further comprising "Means to Store Compatible Graphs" storing [[the]] a compatible graph as a table of function node data, [[that]] which is a table of column function labels of [[said]] each function node and [[the]] data of compatible edges connected to the function node, where the compatible graph is a graph that has nodes of column functions (function nodes), and wherein a pair of function nodes corresponding to column functions compatible each other with an edge or edges (compatible edges), and said method further comprising:

wherein said "Step to Generate Column Functions" ~~in which,~~ further generates column functions corresponding to each edge of nodes at said height of the partition *lev* set in said "Step to Find Dividing Lines" from said node table stored in "Means to Store Node Table", and then generates said function node data labeled by column function labels corresponding to these column functions, and then stores in said "Means to Store Compatible Graphs";

a "Step to Generate Compatible Edges" which selects the pair of compatible column functions from [[the]]\_a set of column functions corresponding to said each function node data, stored in said "Means to Store Compatible Graphs", and then adds compatible edge which connects function node data corresponding to these compatible column functions and these function node, and finally updates the function node data stored in said "Means to Store Compatible Graphs";

a "Step to Generate Cliques" covering all nodes in said compatible graph with [[the]]\_a minimum number of complete subgraphs (cliques) and then generating clique data of the function node set contained in the clique; and

wherein said "Step to Reconstruct Assigned BDD" ~~which~~ further reconstructs said BDD\_for\_CF by making some column functions [[to]]\_into the identically assigned column functions by assigning constants to the *don't care(s)* of the column functions corresponding to each function node contained in the clique data produced by said "Means to Generate Cliques", and updates said node table in said "Means to Store Node Table".

10. (Currently Amended) The Method to Reduce [[the]]\_a Width of a Graph according to Claim 8 wherein:

said "Step to Find Dividing Lines" to said "Step to Reconstruct Assigned BDD" [[is]]\_a performed while changing said height *lev* of the partition sequentially from [[the]]\_a height of [[the]]\_a child node of [[the]]\_a root node in BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table", towards [[the]]\_a lower height.

11. (Currently Amended) A Method for Logic Synthesis which generates look-up tables (LUTs) of [[the]]\_a data for constructing logic circuits corresponding to [[said]]\_a

multiple-output logic function  $F(X)$  from [[the]] a BDD\_for\_CF (Binary Decision Diagram for Characteristic Function) of the multiple-output logic function  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  with input variables  $X=(x_1, \dots, x_n)$  ( $n \in N$ ), in [[the]] a system comprising:

"Means to Store Node Table" storing BDD\_for\_CF representing [[the]] a characteristic function  $\chi(X, Y)$  (where  $Y=(y_0, \dots, y_{m-1})$  ( $m \geq 2$ ,  $m \in N$ ) denotes the output variables of  $F(X)$ ) defined in Equation (11), as [[the]] a node table which is [[the]] a table of [[the]] node data that consists of [[the]] labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , where said multiple-output logic function  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  is a completely specified function, the labels of variables are labels given to [[the]] variables  $z_i$  ( $z_i \in (X \cup Y)$ ) corresponding to [[said]] non-terminal node  $v_i$  in the BDD\_for\_CF, and a pair of edges  $e_0(v_i)$  and  $e_1(v_i)$  that points [[the]] to next transition child node(s) when [[the]] input values of  $z_i$  ( $z_i \in (X \cup Y)$ ) are 0 and 1; and

"Means to Store LUTs" storing said LUTs, and said method further comprising:

a "Step to Find Dividing Lines" setting [[the]] a height of the partition  $lev$  which partitions BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table";

a "Step to Reduce by Shorting" executing, by a processor, shorten-processing [[that]] which is [[the]] a processing to replace [[the]] an edge  $e_c(v_k)$  that points [[the]] to a node  $v_j$  among two edges  $e_0(v_k)$  and  $e_1(v_k)$  of [[the]] a parent node  $v_k$  of the node  $v_j$ , by [[the]] an edge  $e_b(v_j)$  other than [[the]] an edge  $e_a(v_j)$  of the node  $v_j$ , in the case that [[the]] a terminal node related to  $\chi(X, Y)=0$  pointed by the edge  $e_a(v_j)$  of either edge  $e_0(v_j)$  or  $e_1(v_j)$  of



the node  $v_j$ , about the node data of the node  $v_j$  related to [[the]]\_a variable  $y_k(\square Y)$  representing the output and the parent node  $v_k$  of the node  $v_j$ , where the nodes  $v_j$  and  $v_k$  are contained in [[the]]\_a subgraph  $B_0$  including [[the]]\_a root node among the node data of non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning BDD\_for\_CF to [[the]] two subgraphs  $B_0$  and  $B_1$  at [[the]]\_a partition line in said height of the partition  $lev$ ;

a "Step to Measure [[the]]\_a Width of BDDs" which counts [[the]]\_a number of the edges that point [[the]]\_to child nodes of the non-terminal nodes, whose height is smaller than said height of the partition  $lev$ , among the edges which are the non-terminal nodes in BDD\_for\_CF to which said shorten-processing by said "Means to Reduce by Shorting" is applied, and which belong to the non-terminal nodes whose height is larger than said height of the partition  $lev$  (where the edges pointing to the same node is counted as one, and the edges pointing [[the]]\_to\_a constant 0 are ignored), and produces the number of the edges at the partition line in said height of the partition  $lev$  as [[the]]\_a width  $W$ ;

a "Step to Count the Intermediate Variables" counting [[the]]\_a number of [[the]] intermediate variables  $u$  by Equation (12), using the width  $W$ ;

a "Step to form LUT" which generates LUTs from the node data and stores said LUTs in said "Means to Store LUTs", for the non-terminal nodes which belong to the subgraph  $B_0$  including the root node, among the non-terminal nodes stored in said "Means to Store Node Table", in the case of partitioning said height;

a "Step to Reconstruct BDD" which generates a binary tree which has the same number of control inputs as the number of intermediate variables  $u$  which is calculated by said "Means to Compute the Intermediate Variables", and reconstructs the BDD\_for\_CF by

replacing the node data of non-terminal nodes in subgraph  $B_0$  of BDD\_for\_CF stored in said "Means to Store Node Table" with the node data representing said binary tree, and updates the node table stored in said "Means to Store Node Table" by the node data of the non-terminal nodes in said reconstructed BDD\_for\_CF, wherein Equation (11) and Equation (12) are defined as follows:

[Equation 11]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} (y_i \equiv f_i(X)) \quad (11)$$

[Equation 12]

$$u = \lceil \log_2 W \rceil \quad (12)$$

12. (Currently Amended) The Method for Logic Synthesis according to Claim 11, wherein said "Means to Store Node Table" stores the BDD\_for\_CF as a node table, where said BDD\_for\_CF is a graph that represents the characteristic function  $\chi(X, Y)$  (where  $Y=(y_0, \dots, y_{m-1})$  ( $m \geq 2$ ,  $m \in N$ ) denotes the output variables of  $F(X)$ ) defined in Equation (13), with said multiple-output logic function  $F(X)=(f_0(X), \dots, f_{m-1}(X))$  of an incompletely specified function including *don't care* in outputs, said node table is the table of the node data that consists of the labels of variables and pairs of edges  $e_0(v_i)$  and  $e_1(v_i)$ , said labels of variables are labels given to the variables  $z_i$  ( $z_i \in (X \cup Y)$ ) corresponding to said non-terminal node  $v_i$  in the BDD\_for\_CF, and said pair of edges  $e_0(v_i)$  and  $e_1(v_i)$  points to

the next transition child node(s) when the values of  $z_i$  ( $z_i \in (X \cup Y)$ ) are 0 and 1, wherein

Equation (13) is defined as follows:

[Equation 13]

$$\chi(X, Y) = \bigwedge_{i=0}^{m-1} \{\bar{y}_i f_{i,0} \vee y_i f_{i,1} \vee f_{i,d}\} \quad (13)$$

where  $f_{i,0}$ ,  $f_{i,1}$ ,  $f_{i,d}$  are [[the]] OFF-function, [[the]] ON-function and [[the]] DC-function defined in Equation (14), respectively, wherein Equation (14) is defined as follows:

[Equation 14]

$$f_{i,0}(X) = \begin{cases} 1 & (X \in f_i^{-1}(0)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i,1}(X) = \begin{cases} 1 & (X \in f_i^{-1}(1)) \\ 0 & (\text{otherwise}) \end{cases}, f_{i,d}(X) = \begin{cases} 1 & (X \in f_i^{-1}(d)) \\ 0 & (\text{otherwise}) \end{cases} \quad (14)$$

13. (Currently Amended) The Method for Logic Synthesis according to Claim 12 ~~which further~~ reduces the width of BDD\_for\_CF represented by said node table stored in said "Means to Store Node Table", ~~by the Method to Reduce the Width of Graph according to any one of Claims 8 to 10,~~ and then updates said node table stored in said "Means to Store Node Table", and then performs from said "Step to Find Dividing Lines" to said "Step to Reconstruct BDD".

14. (Currently Amended) The Method for Logic Synthesis according to claim 11 wherein; ~~performing the following three steps,~~ after performing from said "Step to Find Dividing Lines" to said "Step to Reconstruct" ~~BDD~~; said method further performs:

a "Step to Decide Ordering of Output Variables" deciding [[the]] order  $\pi$  of elements of said multiple-output logic function  $F(X)$  to minimize [[the]] a value of T represented in

Equation (15), where  $\pi = (\pi[0], \dots, \pi[m-1])$  ( $\pi[l]=j$  represents that  $f_j$  is the  $l$ 'th element) is the order of the logic functions  $f_0(X), \dots, f_{m-1}(X)$  that are elements of said multiple-output logic function  $F(X)$ , and  $\text{supp}(f_j)$  is [[the]] a set of the input variables that influence the logic function  $f_j (\in F(X))$ ;

a "Step to Decide Ordering of all [[the]] Variables" deciding [[the]] an order of [[the]] variables  $y_j (\in Y)$  representing [[the]] outputs and input variables  $x_i (\in X)$  in [[the]] an order  $P$  that satisfies Equation (16); and

a "Step to Generate BDDs" which generates node data of the BDD\_for\_CF according to the order  $P$  decided in said "Means to Decide the Ordering of all the Variables", and then stores in said "Means to Generate BDDs", wherein Equation (15) and Equation (16) are defined as follows:

[Equation 15]

$$T = \sum_{k=0}^{m-1} \left| \bigcup_{l=0}^k \text{supp}(f_{\pi[l]}) \right| \quad (15)$$

[Equation 16]

$$P = \left( \text{supp}(f_{\pi[0]}), y_{\pi[0]}, \text{supp}(f_{\pi[1]}) - \text{supp}(f_{\pi[0]}), y_{\pi[1]}, \text{supp}(f_{\pi[2]}) - \left( \sum_{k=0}^1 \text{supp}(f_{\pi[k]}) \right), y_{\pi[2]}, \dots, \text{supp}(f_{\pi[m-1]}) - \left( \sum_{k=0}^{m-2} \text{supp}(f_{\pi[k]}) \right), y_{\pi[m-1]} \right) \quad (16)$$

15. (Currently Amended) A non-transitory computer program that implements readable medium embedded thereon a computer program, which, when executed by a

computer, causes the computer to implement the Method to Reduce the Width of Graph according to claim 8.

16. (Currently Amended) A non-transitory computer program that implements readable medium embedded thereon a computer program, which, when executed by a computer, causes the computer to implement the Method for Logic Synthesis according to claim 11.

17. (Canceled)